

Uek223 Doku Gruppe 5

Dokumentinformationen

Dateiname: uek223 Dokumentation
Speicherdatum: 23.02.2024

Autoreninformationen

Autor: ÖMER BURAK ADANIR, YAN PISCHAN, LORENA VENNEMAN

Inhaltsverzeichnis

1	Einleitung	3
1.1	Sinn und Zweck	3
1.2	Das Team	3
1.3	Ausgangslage	3
2	Auftrag	3
2.1	3
2.2	Arbeitseinteilung	4
3	Systemarchitekturen	4
3.1	Sequence Diagram	4
3.2	Use-Cases	5
3.3	Entity Relationship Diagram.....	6
4	Navigation.....	7
5	Testing	7
5.1	Postman (Back-End).....	7
5.2	Cypress (Front-End)	7

Änderungsgeschichte

Version	Datum	Autor	Details
1.0	23.02.2024		Dokument erstellt

Tabelle 1 Versionen

1 Einleitung

1.1 Sinn und Zweck

Das vorliegende Dokument wurde im Rahmen des überbetrieblichen Kurses zu Multiuser Applikation entwickeln als Dokumentation geschrieben.

1.2 Das Team

1.3 Ausgangslage

Im Rahmen des aktuellen Projektauftrags wurde uns die Aufgabe zugeteilt, bei einer rudimentären Full-Stack Social-Media-Anwendung einige neue Funktionalitäten hinzuzufügen. Eine Erweiterung welche Usern ermöglicht eine «Image Gallery» zu haben.

2 Auftrag

2.1

Das Ziel dieses Projektes ist, wie bereits erwähnt, die Entwicklung einer Full-Stack Komponente durch React, für das Frontend, Spring Boot für das Backend und PostgreSQL für die Datenerhaltung zu umsetzen. Die folgenden Anforderungen sind spezifisch für unser Projekt festgelegt worden:

1. Erstellen einer Image-Post Model, welche Informationen über ein Foto enthält, welches von einem User geteilt wird (URL, Author, Beschreib, Likes (welche User haben dieses Bild geliked)).
2. Jeder User kann mehrere Posts erstellen und die Posts gehören jeweils einer Person.
3. Endpoints, welche die CRUD-Operationen an Listeneinträgen durchführen.
4. Endpoint welche alle Posts eines Users ausgibt (nur für eingeloggte User zugänglich)
5. Users können ihre eigenen Posts bearbeiten oder Admins alle

Nebst den Gruppenspezifischen Anforderungen haben wir auch funktionale Anforderungen für die User Rollen und Privilegien, Frontend und Security. Aber auch nicht funktionale bei welchem Wert auf die Implementation, Testing, Multiuserfähigkeit und Dokumentation Wert gelegt wird. Näheres kann in diesem [Dokument](#) gelesen werden.

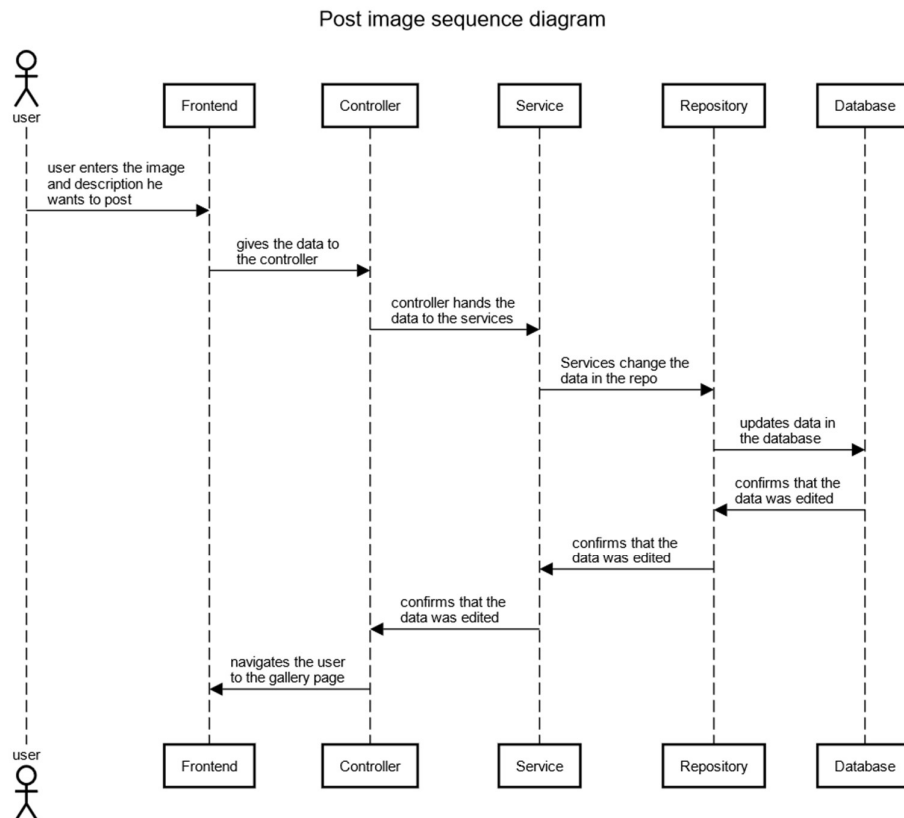
2.2 Arbeitseinteilung

Zu Beginn des Projekts haben wir als Team gemeinsam daran gearbeitet, das bestehende Programm welches wir erhalten haben, zum Laufen zu bringen. Danach haben wir uns Aufgabenbereiche zugeteilt, um effizienter arbeiten zu können. Dabei haben wir eine hauptsächlich eine Teilung zwischen Front-End und Back-End Entwicklung gemacht. Ich, Ömer, übernahm zusammen mit Lorena die Verantwortung für das Back-End, während Yan sich hauptsächlich auf das Front-End konzentrierte.

Parallel zu meiner Aufgabe im Entwicklungsbereich, habe ich mit der Dokumentation begonnen, habe mich mit der Datenbank beschäftigt und Tests mit Postman durchgeführt. Mit dem frühen Testing ermöglichten wir es sicherzustellen, dass alle Funktionen wie erwartet funktionierten und wir mögliche Fehler frühzeitig entdecken könnten. Nach der Fertigstellung des Back-Ends wechselte Lorena zur Unterstützung ins Front-End Team zu Yan.

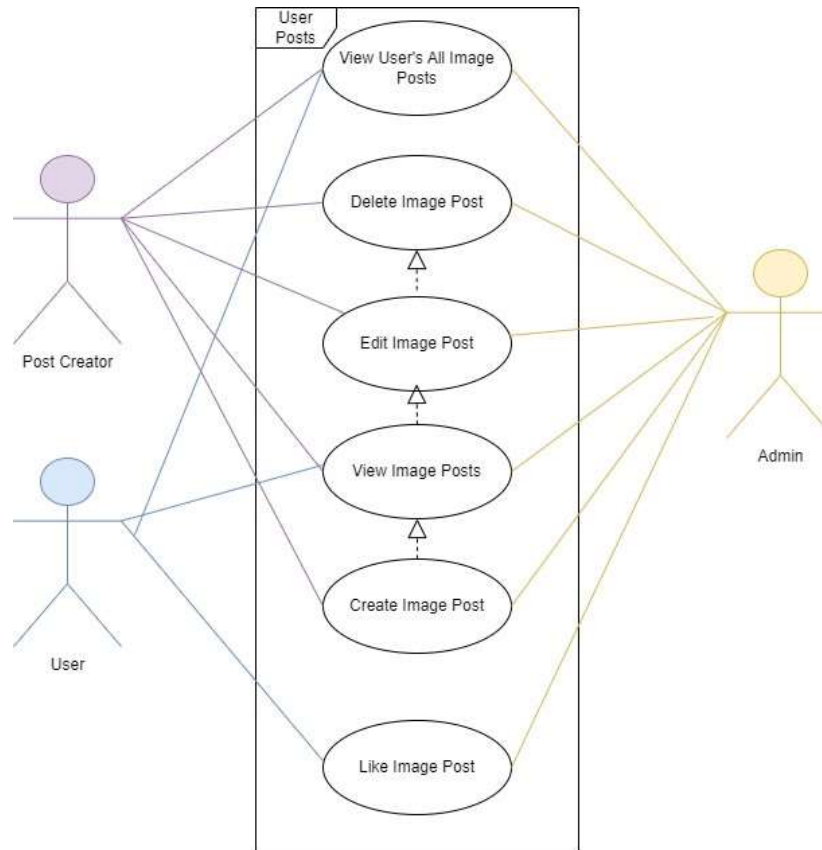
3 Systemarchitekuren

3.1 Sequence Diagram



3.2 Use-Cases

3.2.1 Use-Case diagram



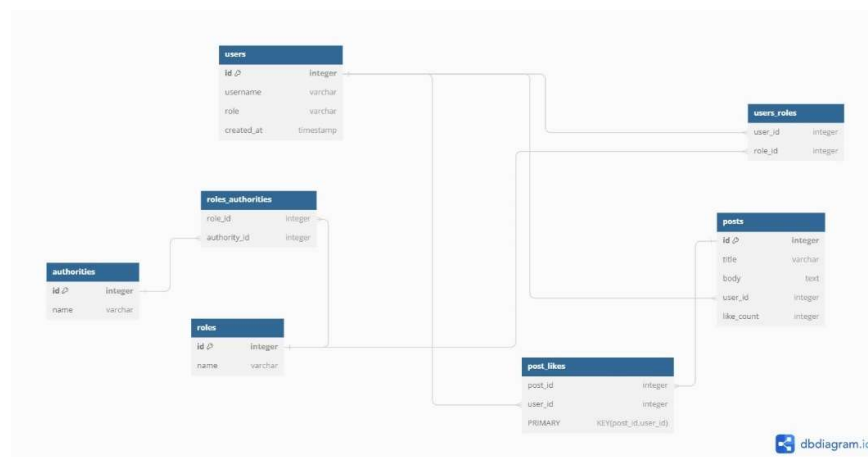
3.2.2 Use-Case Definition

Actor(s)	User
Beschreibung:	Ein Benutzer kann ein Image Post erstellen mit einem Bild Link, Beschreibung und Like Fähigkeit von Usern nach dem Registrieren erstellen. Kann eigene Posts bearbeiten und löschen.
Preconditions/Voraussetzungen	<ul style="list-style-type: none"> User ist registriert und eingeloggt
Postconditions/Nachbedingungen	<ul style="list-style-type: none"> User hat Post erstellt oder bearbeitet
Normaler Ablauf	<ul style="list-style-type: none"> User loggt ein User erstellt Post Image Post Daten werden gespeichert und angezeigt. User kann (eigenes) Post liken
Alternative Abläufe	2. User bearbeitet Post und macht ein Update davon 2. User löscht Post
Ausnahmen	Keine

Actor(s)	Admin
Beschreibung:	Ein Admin kann ein Image Post erstellen mit einem Bild Link, Beschreibung und Like Fähigkeit von Usern nach dem Registrieren erstellen. Kann aber auch Posts anderer User bearbeiten und löschen.
Preconditions/Voraussetzungen	<ul style="list-style-type: none"> • Admin ist registriert und eingeloggt
Postconditions/Nachbedingungen	<ul style="list-style-type: none"> • Admin hat Post erstellt oder bearbeitet • Admin hat Users Post bearbeitet oder gelöscht
Normaler Ablauf	<ul style="list-style-type: none"> • Admin loggt ein • Admin erstellt Post • Image Post Daten werden gespeichert und angezeigt. • Admin kann (eigenes) Post liken
Alternative Abläufe	2. Admin bearbeitet Post und macht ein Update davon 2. Admin löscht Post 2. Admin bearbeitet Users Post 2. Admin löscht Users Post
Ausnahmen	Keine

3.3 Entity Relationship Diagram

Dies ist unser ERD, wir haben dieses Diagramm im Rahmen der UML Präsentationsreihe erstellt.

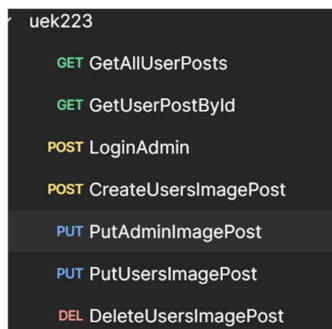


4 Navigation

5 Testing

Bislang konnten wir nur ein Paar Tests mithilfe von Postman fürs Back-End durchführen, allerdings planen wir mit Cypress das Front-End zu testen und wenige Unit-Tests zu schreiben, um die Funktionalität unseres Endproduktes durchzuführen.

5.1 Postman (Back-End)



Dies sind unsere bisherigen Tests mit Postman. Wir haben es für das Back-End ausschliesslich auf Component Niveau verwendet.

5.2 Cypress (Front-End)

Test Level:	End-to-End
Tool	Cypress
<ul style="list-style-type: none">Cypress wird oft für Tests verwendet, weil es erstens eine schnelle und zuverlässige Testausführung für End-to-End-Tests bietet, indem es direkt im Browser läuft. Zweitens bietet es eine einfache und intuitive API, die das Schreiben von Tests erleichtert und eine hervorragende Dokumentation und Community-Unterstützung hat.Schnelle und simple End-to-End Testausführung, läuft direkt im BrowserNimmt Test auf und kann Einblicke verleihen und automatisiert unter anderem auch User interactions	

6 Benötigte Tools

- IntelliJ IDEA
- Visual Studio Code
- Docker Desktop
- Postman
- Draw.io
- Dbdiagram.io
- Cypress

7 GitHub

Backend - https://github.com/LorenaVennemann/uek223_Backend

Frontend - https://github.com/LorenaVennemann/uek223_Frontend